

# SPANISH RECOGNIZER OF CONTINUOUSLY SPELLED NAMES OVER THE TELEPHONE

*Rubén San-Segundo, José Colás, Ricardo de Córdoba, José M. Pardo*  
Grupo de Tecnología del Habla. Departamento de Ingeniería Electrónica. UPM  
E.T.S.I. Telecomunicación. Ciudad Universitaria s/n, 28040 Madrid, Spain.  
[lapiz@die.upm.es](mailto:lapiz@die.upm.es)

## ABSTRACT

*In this paper we present a hypothesis-verification approach for a Spanish Recognizer of continuously spelled names over the telephone. We give a detailed description of the spelling task for Spanish where the most confusable letter sets are described. We introduce a new HMM topology with contextual silences incorporated into the letter model to deal with pauses between letters, increasing the Letter Accuracy by 6.6 points compared with a single silence model approach. For the final configuration of the hypothesis step we obtain a Letter Accuracy of 88.1% and a Name Recognition Rate of 94.2% for a 1,000 names dictionary. In this configuration, we also use noise models for reducing letter insertions, and a Letter Graph to incorporate N-gram language models and to calculate the N-best letter sequences. In the verification step, we consider the M-Best candidates provided by the hypothesis step. We evaluate the whole system for different dictionaries, obtaining more than 90.0% Name Recognition Rate for a 10,000 names dictionary. Finally, we demonstrate the utility of incorporating a Spelled Name Recognizer in a Directory Assistance Service over the telephone increasing the percentage of calls automatically serviced from 39.4% to 58.7%.*

## 1. Introduction

Automatic speech recognition of names from its spelling is an important sub-task for many applications such as directory assistance (Lehtinen, 2000) and (Schrâmm, 2000), or identification of city names for travel services (Lamel, 2000). Natural spelling implies the recognition of connected letters. This is a difficult task, especially over the telephone, because of the confusable letters contained in the alphabet, the distortions introduced by the telephone channel and the variability due to an arbitrary telephone handset.

### 1.1 The Spelling Task for Spanish.

The performance of a recognition system depends not only on the size or perplexity of the vocabulary, but also on the degree of similarity between the words in the vocabulary. In Table 1, we present the transcriptions of Spanish-letter pronunciations using the International Phonetic Alphabet (IPA).

Table 1.  
Spanish-Letter transcriptions.

| Spanish-Letter transcriptions (IPA) |    |          |       |           |      |          |      |          |           |
|-------------------------------------|----|----------|-------|-----------|------|----------|------|----------|-----------|
| <b>A</b>                            | a  | <b>F</b> | efe   | <b>L</b>  | 'ele | <b>P</b> | pe   | <b>V</b> | 'ube      |
| <b>B</b>                            | be | <b>G</b> | ge    | <b>LL</b> | 'eëe | <b>Q</b> | ku   | <b>W</b> | ube'doble |
| <b>C</b>                            | θe | <b>H</b> | 'ate  | <b>M</b>  | 'eme | <b>R</b> | 'ere | <b>X</b> | 'ekis     |
| <b>Ch</b>                           | te | <b>I</b> | i     | <b>N</b>  | 'ene | <b>S</b> | 'ese | <b>Y</b> | 'igrjega  |
| <b>D</b>                            | de | <b>J</b> | 'xota | <b>Ñ</b>  | 'e e | <b>T</b> | te   | <b>Z</b> | 'θeta     |
| <b>E</b>                            | e  | <b>K</b> | ka    | <b>O</b>  | o    | <b>U</b> | u    |          |           |

In English, the main difficulty in the spelling recognition task lies in the recognition of the E-set = {B, C, D, E, G, P, T, V, Z} (Loizou, 1996). Looking at Table 1, we can identify the E-set for Spanish = {B, C, CH, D, E, G, P, T}. In this set, the problems of confusion between letters are similar to English. In Spanish, we have to consider another highly confusable letter set, the ExE-set = {F, L, LL, M, N, Ñ, R, S}. In this set, all the letter transcriptions have three phonemes with the structure 'e \_ e. These letters have only one phoneme different (the central phoneme), so the acoustic differences (as in the E-set) are minimal. The most difficult letters to discriminate in the ExE-set are the following:

- *Letters F and S:* in both cases the central phoneme is voiceless and fricative, the difference is the articulation point: /f/ is labiodental and /s/ alveolar.
- *Letters L and LL:* the central phoneme is voiced and lateral in both cases, but the articulation point is different: /l/ is alveolar and /ʎ/ palatal.
- *Letters M, N and Ñ:* the central phoneme is voiced and nasal, but the articulation point is different: /m/ is bilabial, /n/ alveolar and /ɲ/ palatal. The main difference lies in the vowel to nasal articulation.

Outside these sets, there are two pairs of letters with high degree of similarity: letters (K, A) and (Q, U). In both cases the difference is the voiced stop consonant /k/. This phone is only a small portion of the whole letter. Sometimes, the duration of this phone (around 30 ms, calculated from 200 training files) is not enough for discrimination.

When we work with continuous speech, another source of recognition errors is the co-articulation between words. This effect is more dangerous when the words of the vocabulary are shorter and their pronunciations are similar, as in our case. In Fig. 1, we can see an example of recognition error because of the co-articulation effect.

|                             |        |        |        |   |        |
|-----------------------------|--------|--------|--------|---|--------|
| NAME SPELLED:               | R      | U      | B      | E | N      |
| TRANSCRIPTION:              | 'e R e | u      | b e    | e | 'e n e |
| TRANSCRIPTION RECOGNIZED:   | 'e R e | 'u b e | 'e n e |   |        |
| LETTER SEQUENCE RECOGNIZED: | R      | V      | N      |   |        |

Fig. 1. Example of recognition error because of the co-articulation effect.

In this example, the pronunciations of the letters U and B have been joined to form the letter V, and the letter E was included in the N pronunciation. The recognized letter sequence in this case, is quite different from the name spelled.

Other important aspect that we have to keep in mind is the alternative pronunciations. In Table 1, we presented the standard letter pronunciations, and now, in Table 2, we show the most common alternative letter pronunciations for Spanish. To deal with this, we have to train specific acoustic models for them.

Table 2.  
Most common second letter pronunciations for Spanish.

| Second letter pronunciations |                   |
|------------------------------|-------------------|
| <b>CH</b>                    | θ e 'a t e        |
| <b>I</b>                     | 'i l a t 'i n a   |
| <b>LL</b>                    | 'e l e 'd o b l e |
|                              | 'd o b l e 'e l e |
| <b>R</b>                     | 'e r e            |
| <b>W</b>                     | 'd o b l e 'u b e |

## 1.2 Previous Research.

Early systems for isolated letter recognition used template matching with dynamic time warping (DTW) (Cole, 1986) or a combination of template matching with feature-based speech knowledge (Junqua, 1991). In the past decade hidden Markov models (HMM's) (Brown, 1987), (Euler, 1990), (Junqua, 1997) and artificial neural networks (ANN's) (Cole, 1991b) (Roginski, 1991), (Hild, 1993), adequately adapted to the task, were shown to perform quite well. While HMM-based systems generally focus on state-tying to enhance the discrimination between confusable letters, ANN-based systems perform discrimination based on a segmentation and/or a pre-classification given by a conventional approach (DTW or HMM).

In the 90's, the problem of automatic speech recognition of continuously spelled names over the telephone has attracted a lot of interest (Jouvet, 1993a), (Jouvet, 1993b), (Kaspar, 1995), (Loizou, 1995). This was partially due to the availability of several alphabet corpora developed and distributed by the Center for Spoken Language Understanding at OGI. OGI used these corpora to develop an accurate letter recognition system called the English Alphabet Recognizer (EAR) (Fanty, 1990). The research was later extended to alphabet recognition of telephone speech using a slightly different approach (Cole, 1991b), (Junqua, 1997). Several methods made use of N-best strategies followed by postprocessors to reorder N-best candidates.

Currently, spelled name recognition systems are being widely used as a fall back strategy (Bauer, 1999) and for rejection of incorrect data (Jouvet, 1999). In these situations a high level of accuracy is required for these systems. Because of this, approaches based on several recognition stages are widely used (Mitchell, 1999), (Junqua, 1997) and long range language models are incorporated (Thiele, 2000). In (Hanel, 2000) statistics on recognized letter sequences are used to detect the end of spelling in continuously spelled names.

In this paper, we use a hypothesis-verification approach for developing the first spelled name recognizer for Spanish with results comparable to systems developed for other languages and we demonstrate the utility of incorporating this recognizer in a Directory Assistance service. We also propose a new HMM topology with contextual silences to deal with pauses between letters that gives significant improved performance over previous systems. The paper is organised as follows. Section 2 introduces the system overview, and a high level description of the main modules (hypothesis and verification). In Section 3, we describe the database used to train and evaluate our alphabet recognizer. Section 4 presents a new HMM topology with contextual silences and the results of introducing noise models, N-gram language models, N-best technique and the letter-graph approach in the hypothesis stage. Section 5 describes the verification stage and the analysis of the number of candidates passed from the hypothesis to the verification stage. The field evaluation is shown in the Section 6. Finally, in Section 7 we review the conclusions of this work.

## 2. System overview

The system proposed is based on a hypothesis-verification approach. Our recognition strategy consists of two steps: in the hypothesis step, we obtain N-best sequences of letters given acoustic HMMs of the letters and then, we compare these sequences with all the names in the dictionary using a dynamic programming algorithm to obtain the M-best similar names. These names are passed to the verification step. In this stage, a dynamic grammar is built with the M-best names and the HMM recognizer is rerun with this highly constrained grammar. In Fig. 2, we can see the block diagram of the system.

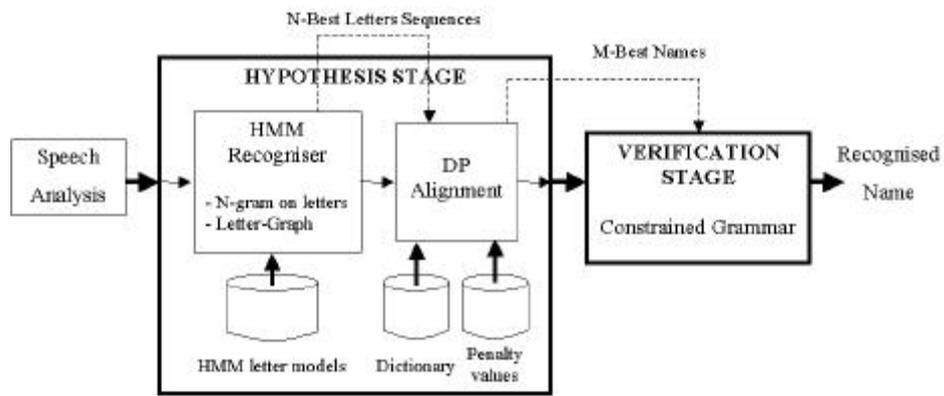


Fig. 2. Block Diagram of the system.

Our approach is similar to (Junqua, 1997) but we introduce a new HMM topology with contextual silences and we generate a letter graph similar to the word-graph proposed in (Ney, 1994), (Ney, 1999) to incorporate N-gram language models and to obtain N-best letter sequences.

### 2.1 Speech Analysis.

We use a 10 ms frame shift and a 25 ms analysis window. In the experiments, the static cepstral coefficients, the local energy, the first derivative of the energy and the first derivative of the static cepstral coefficients are considered to form the speech parametric representation (a total of 22 coefficients). We use the RASTA-PLP (Hermansky, 1991) parameterisation as proposed in (Junqua, 1997) where we can see a detailed description of the Front-End analysis.

### 2.2 Hypothesis Stage.

In the HMM recognizer we use continuous density HMMs (C-HMMs) with a number of states proportional to the length of each letter. We have a different model for each letter. We use C-HMMs because they are the most powerful HMMs and can deal better with the high level of confusion in the recognition vocabulary. The shortest model has 9 states and it is associated to the vowel letter I, the longest one has 48 and it is associated to letter W. The number of mixtures per state is proportional to the amount of data used for training. We consider a minimum number of three mixtures and a maximum of nine. The HMM training has been carried out by a standard training procedure. We use the Viterbi algorithm for assigning each frame to a state and the Expectation-Maximisation (EM) re-estimation for calculating iteratively the mixtures in each state. We obtain an initial set of mixtures per state using the LBG algorithm (Linde, 80).

In this stage we propose a new HMM topology with contextual silences, we analyze the introduction of noise models in the search space and we consider several N-gram language models (bigram and trigram) and N-best letter sequences in the one-pass algorithm. The language models have been generated with the name directory of the task. We introduce a Letter Graph Generation to incorporate the 3-gram language model in the search space and to calculate N-best letter sequences.

Obtained the N-best letter sequences, our target is to get the M-best names from the dictionary. The N letter sequences are sequentially compared to all the dictionary names using a Dynamic Programming (DP) algorithm. The DP algorithm applies different penalties for possible substitutions, deletions and insertions in the letter sequence. The best path along the search space is computed and the total alignment cost is calculated. Given a letter sequence, the dictionary names with lowest alignment cost are selected. For training the penalties, the letter sequence is compared with the spelled name and the number of substitutions, insertions and deletions are then counted (Fissore, 1989). The penalties are calculated as the logarithm of the inverse probability obtained from the counted events. This process is repeated iteratively until

the penalty values stop changing significantly. As initial values we have considered a penalty 1 for insertions and deletions, 2 for substitutions and 0 for a correct letter. The different penalty values are adapted every time the first step HMM recognizer is modified.

### 2.3 Verification Stage

In this stage, we use the M-best candidate names to build a dynamic grammar and the HMM recognizer is invoked again with this constrained grammar. In this step, the time consumption is low because the number of names considered is small (in section 5 we present an analysis to choose this number). Another reason is that we use the same HMMs as in the hypothesis stage, so the state distribution probabilities, calculated in this stage, are stored for the verification stage. In our experiments, we used the same HMMs for both steps but more detailed models or different recognition parameters can be used in the verification stage. This stage is similar to the 4<sup>th</sup> pass proposed in (Junqua, 1995).

## 3. Data base

The database used for the experiments is the SpeechDat database (Moreno, 1997). This database was recorded from the Spanish telephone network using 1000 speakers. Each speaker was asked to spell a city name, a proper name and a random letter sequence (this guarantees a minimum number of training examples for each letter) providing around 22,800 letters in 3,000 audio files. The random sequences of letters are used only for HMM training. From the city and proper names audio files we have randomly taken 600 files (300 speakers) for two development sets, the first one (300 files, 150 speakers) for adapting the DP algorithm penalties and the second (300 files, 150 speakers) for development. We also take 300 files (150 speakers) for final testing, leaving the rest (2100 audio files) for HMM training<sup>1</sup>. We have repeated it six times providing a 6-Round Robin training to verify the results. In this paper, we present the average results of these experiments.

We consider several dictionaries of different sizes (1000, 5000 and 10,000 city and proper names) obtained by randomly extracting from the Spanish city and proper name directory. The city and proper names spelled in the database are included in every dictionary. The average confusion for the dictionaries is 0.2, 0.5 and 0.9 respectively. This measure is calculated as the average number of name pairs from the dictionary that differ only by one letter substitution. These values are a measure of dictionary confusion (Cole, 1991a), (Junqua, 1997) and provide an idea on the difficulty of the recognition task. In the third dictionary (10,000 names), there were 9,038 pairs of names that differ only by one letter substitution. This corresponds to an average of 0.9 confusions per name.

We will report the percentage of Substitutions, Deletions and Insertions, the Letter Accuracy and the Name Accuracy obtained, to evaluate the different alternatives proposed for the HMM recognizer in the hypothesis step. We consider the *Name Accuracy* as the percentage of cases where the letter sequence recognized matches exactly with the name spelled. To analyze the hypothesis stage and the whole system (hypothesis + verification), we will show the Name Recognition Rate obtained after the hypothesis stage and after the whole system. The confidence interval of the results at 95% is generally less than  $\pm 0.7\%$  for the Letter Accuracy and  $\pm 1.4\%$  for the Name Recognition Rate. We calculate the confidence interval using the equation 1 proposed in (Weiss, 1993):

$$\frac{band}{2} = z_{\alpha/2} \sqrt{\frac{p(100-p)}{n}} \quad (1)$$

---

<sup>1</sup> One can argue that this special division of the database is not totally speaker independent because there is one sentence per testing speaker included in the training database. We have done some of the experiments excluding those sentences and its influence is negligible (see Appendix A).

where  $p$  is the percentage sample proportion (number of successes divided by sample size  $\times 100$ ),  $n$  the sample size and  $z_{\alpha/2} = 1.96$  for a 95% confidence interval. Any recognition or error rate is translated into the band  $[p - \text{band}/2, p + \text{band}/2]$  with a confidence of 95%.

All the experiments have been run using a Pentium II 350 Mhz with RAM of 128 Mb, so all the processing time results provided refer to this computer. These results are given in xRealTime units (xRT) as in (Ravishankar, 1996). 1xRT is the average time spent to pronounce the spelled name.

#### 4. Hypothesis stage

As we showed in Fig. 2, the hypothesis stage is made up of two modules: the HMM recognizer (letter sequence generator) and the DP alignment with the dictionary. In this section, we are focusing on the HMM recognizer but we will present results for both modules. For the experiments in this section, we have considered the 1,000 name dictionary.

##### 4.1 Baseline system.

As a baseline HMM recognizer we have considered a one-pass algorithm with 35 letter models (standard and second pronunciations) and a single silence model. In order to deal with different silence durations, we have introduced a backward transition on the silence model from the last state to the first silence state to allow the concatenation of several ones. During HMM training, it has been necessary to incorporate this transition in order to train its probability. We present the results in table 3.

Table 3.

Results for the baseline hypothesis stage: Percentage of Substitutions (Sub), Insertions (Ins) and Deletions (Del), Letter Accuracy (LA), Name Accuracy (NA) and Name Recognition Rate (NRR) with the Processing Time (PT) consumed by the whole hypothesis stage (considering the 1,000 names dictionary).

| Baseline system |        |        |                  |       |        |         |
|-----------------|--------|--------|------------------|-------|--------|---------|
| HMM recognizer  |        |        | Hypothesis stage |       |        |         |
| Sub(%)          | Ins(%) | Del(%) | LA(%)            | NA(%) | NRR(%) | PT(xRT) |
| 20.2            | 6.5    | 3.7    | 69.6             | 21.3  | 83.4   | 0.9     |

As we can see, the percentage of substitutions is quite important because of the high level of confusion between letters. Another important fact is the low Name Accuracy. It is quite difficult to recognize a letter sequence that matches the name spelled perfectly. Because of this, the DP Alignment module is needed to increase the Name Recognition Rate. In our case, this increment has been considerable, obtaining a 83.4% Name Recognition Rate for the 1,000 names dictionary.

##### 4.2 HMM topology with contextual silences incorporated (HMM-CS).

In continuous spelled names the pauses between letters can change their length a lot depending considerably on the person's habits. To deal with this characteristic we have considered a new HMM topology with contextual silences. In this topology, each letter model has two three-state silence models associated with it, one to model the possible previous silence and the other, to model the possible posterior silence. In Fig.3 we can see an example for the letter A.

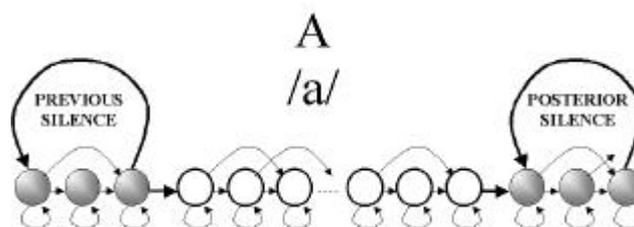


Fig. 3. New HMM topology with contextual silences incorporated.

As in the baseline experiment, we have considered a backward transition on the silence models from the last state to the first silence state to model different silence durations. Considering 35 letter models, we need to train 70 three-state silence models, 35 previous and 35 posterior models. We have been able to estimate the silence models properly because files used for HMMs training, contain a high variability of silence durations between letters.

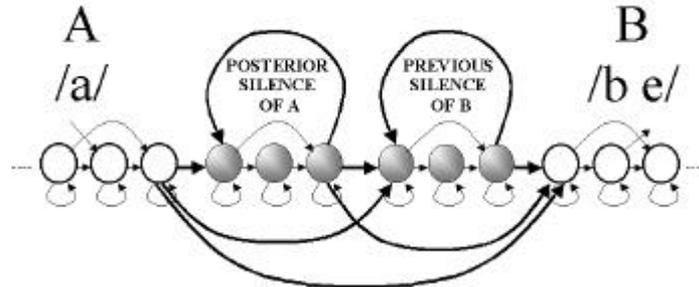


Fig. 4. Transitions between letters with contextual silence models.

In Fig. 4, we can see the new transitions between letter models that appear in the one-pass algorithm when we consider the contextual silence models. We present the results of this topology in table 4. As we can see, incorporating contextual silence models in the letter model has increased the processing time, compared with one silence model approach, but the Letter Accuracy has improved 6.6 points (from 69.6% to 76.2%) obtaining a 89.1% Name Recognition Rate for the hypothesis stage. In Spanish, there is an almost direct correspondence between graphemes and phonemes so the number of homophones is lower than in English or German. Because of this, people are not used to spelling for clarification and they produce a lot of pauses between letters with high duration variability. We have evaluated these pauses in 100 files from the training set and we have obtained an average pause duration of 0.26 s with a Typical Deviation of 0.24 s (Variance 0.06). The contextual silences deal better with these effects obtaining a higher accuracy in the letter sequence segmentation. We think this is an explanation for the improvements due to the silence models.

#### 4.3 Noise models.

Apart from the variability in pauses, other frequent effects produced by Spanish speakers when they spell are false beginnings, doubts, filled pauses and mistakes. Therefore, on the telephone, the speech input may be contaminated with various ambient noises. To deal with these irrelevant sounds we have included 4 noise models in the search space:

- *[fil]*: *Filled pause*. This model corresponds to filled pause sounds. Examples of filled pauses: uh, um, er, ah, mm.
- *[spk]*: *Speaker noise*. All kinds of sounds and noises made by the calling speaker that are not part of the prompted text, e.g. lip smack, cough, grunt, throat clear, tongue click, loud breath, laugh, loud sigh.
- *[sta]*: *Stationary noise*. This category contains background noise that is not intermittent and has a more or less stable amplitude spectrum. Examples: car noise, road noise, channel noise, GSM noise, voice babble (cocktail-party noise), public place background noise, street noise.
- *[int]*: *Intermittent noise*. This model corresponds to noises of an intermittent nature or whose spectrum changes over time. Examples: music, background speech, baby crying, phone ringing, door slam, door bell, paper rustle, cross talk.

Training these noise models has been possible because the SpeechDat database contains their transcriptions within the audio files. We analyzed the training set and we observed that

65.1% of the files contain one or more of these noises. The results obtained with these noise models included in the search space, are presented in Table 4.

Table 4.

Results for the HMM topology with contextual silences (HMM-CS) and the 4 noise models (N-HMMs) incorporated in the search space: Percentage of Substitutions (Sub), Insertions (Ins) and Deletions (Del), Letter Accuracy (LA), Name Accuracy (NA), Name Recognition Rate (NRR) and Processing Time (PT), consumed by the whole hypothesis stage (considering the 1,000 dictionary), are also shown.

|                     | HMM recognizer |        |        |       |       | Hypothesis stage |         |
|---------------------|----------------|--------|--------|-------|-------|------------------|---------|
|                     | Sub(%)         | Ins(%) | Del(%) | LA(%) | NA(%) | NRR(%)           | PT(xRT) |
| Baseline            | 20.2           | 6.5    | 3.7    | 69.6  | 21.3  | 83.4             | 0.9     |
| HMM-CS              | 17.3           | 4.2    | 2.3    | 76.2  | 27.8  | 89.1             | 1.2     |
| HMM-CS + Noise HMMs | 15.9           | 1.1    | 2.3    | 80.7  | 34.3  | 92.0             | 1.2     |

The noise models provide a good improvement in the system. With almost the same processing time the noise models reduce the substitutions 1.4 and the insertions 3.1 points (from 4.2% to 1.1%), increasing 2.9 points the Name Recognition Rate.

#### 4.4 Language Models.

When the spelled name belongs to a finite known list (dictionary) as in our case, this list can provide very useful information that can be used in several ways (Hild, 1997). One way of considering this information in the HMM recognizer is defining N-gram language models and including them in the search space. We calculate a 2-gram and 3-gram language models using the 1,000 name dictionary. The probabilities were smoothed using a minimum probability value optimized over the first development set described in section 3. Incorporating the 2-gram language model is rather easy because it is not necessary to change the search space. We only need to consider the LM probability when we analyze a possible transition between two letters. In the case of 3-gram LM, it is necessary to change the search space. We have to duplicate every letter model as many times as letters that can precede it. In Fig. 5, we present the search space considered to include the 3-gram LM. Nodes (X,Y) represent acoustic models of the letter Y with tracking information coming from the letter X. We have also considered (INI, Y) nodes for taking into account first letter of sequences. They represent acoustic models of the letter Y with tracking information coming from the Initial Node (INI). The INI only can transit to these nodes.

One problem we found in this structure was the way of including the noise models in the search space. The solution is to duplicate the noise HMMs as many times as nodes we have in the structure. This solution increases the space size a lot and makes the search very slow. Following the idea proposed in (Kuroiwa, 1999), we analyzed the noise distribution throughout the utterance. We studied the training set and we found that 67.3% of the noises appear at the beginning of the utterance, 26.2% at the end and only the 6.5% appear between letters. So, we decided to incorporate the noise models only in the Initial and End nodes (Fig. 5). This way, we do not increase the space size so much but we can detect a significant amount of noise.

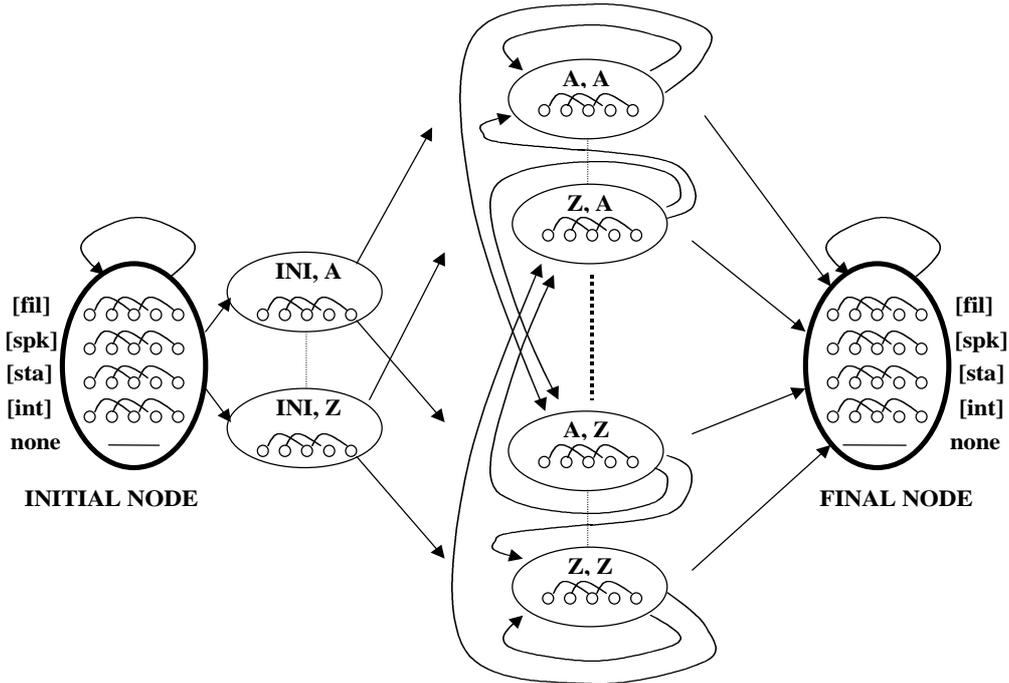


Fig. 5. Noise models incorporation into the 3-gram LM search space.

A recognized letter sequence can start with a noise or not: to represent this possibility, we have introduced a line ("none") in the Initial Node. In the same way, the letter sequence can finish with a noise or with a letter. We have also introduced a line ("none") in the Final Node. All the nodes represented can transit to the Final Node for finishing the letter sequence at any moment. In this figure, we have not represented the silence models because they are included in the letter or noise models (section 4.2).

As we can see in table 5, the 3-gram LM is more powerful than the 2-gram. We obtain a significant improvement in the Letter and Name Accuracy with a relative error reduction of 40.9% and 38.7% respectively. This reduction has been possible thanks to the grammar information rather than to the acoustics models. In Fig. 5, we only represent one HMM per node but in the real case, it is necessary consider as many HMMs as different pronunciations we have for each letter.

Table 5.

Letter Accuracy (LA), Name Accuracy (NA), Name Recognition Rate (NRR) and Processing Time (PT) for the Baseline system, considering contextual silences (HMM-CS) and noise models (N-HMMs), and including 2-gram/3-gram LM with N-HMMs.

|                          | LA (%) | NA (%) | NRR(%) | PT(xRT) |
|--------------------------|--------|--------|--------|---------|
| Baseline                 | 69.6   | 21.3   | 83.4   | 0.9     |
| HMM-CS                   | 76.2   | 27.8   | 89.1   | 1.2     |
| HMM-CS + N-HMMs          | 80.3   | 33.8   | 91.7   | 1.2     |
| HMM-CS + N-HMMs + 2-gram | 81.4   | 35.4   | 92.3   | 1.3     |
| HMM-CS + N-HMMs + 3-gram | 89.0   | 60.4   | 93.2   | 3.8     |

This increment in the Letter Accuracy produces a small improvement in the Name Recognition Rate of the hypothesis stage. The reason is that we have reduced more than 40% of the errors in the letter sequence so we have less data to train the insertion, deletion and substitution penalties of the DP algorithm in this case. Another reason is that the penalties used do not consider contextual information, i.e. we do not train different penalties depending on the

context. When we work with LMs, an error made in a letter can produce new errors in the preceding or following letters so context independent penalties do not correctly model these mistakes. Let us look at the following example in Fig. 6.

|           | <b>Without LM</b> | <b>With LM</b> |
|-----------|-------------------|----------------|
| Reference | R U B E N         | R U B E N      |
| Case 1    | F U P E N         | R U P I N      |
| Case 2    | R U D E N         | R U D N        |

Fig. 6. Different errors because of LM introduction.

Without LM the letter B is substituted by P and D respectively, but the letter E has been recognized correctly. When we introduce the LM, substitutions in the letter B can produce different errors in the contextual letter E. Training penalties depending on the preceding or following letters have not been possible because we do not have enough data to train them.

#### 4.5. Letter Graph.

The main idea of the word graph for continuous speech recognition proposed by Hermann Ney in (Ney, 1994), (Ney, 1999) is to come up with word alternatives in regions of the speech signal, where the ambiguity in the acoustic recognition is high. If we can build a letter graph, similar to a word graph for continuous speech recognition, it would be possible to obtain the N-best letter sequences and to incorporate N-gram language models with low time consumption.

For letter graph generation it is necessary to keep track of letter sequence hypotheses whose scores are very close to the locally optimal hypothesis, but that do not survive due to the recombination process in the one-pass algorithm. The basic idea is to represent all these sequences by a graph, in which each node represents a letter hypothesis. Each letter sequence contained in the graph should be close (in terms of scoring) to the single best sentence produced by the one-pass algorithm. For the letter graph construction, it is necessary to consider the following steps:

- At each time  $t$ , for each letter  $L_1$  we consider all possible predecessors to it and select the most probable letter transitions  $(L_i, L_1)$ . H. Ney in (Ney, 1999) proposes the use of the beam search strategy to obtain a limited number of predecessors. In our case, we do not use the beam search strategy because the recognition vocabulary is small (35 letter + 4 noise models), so we define a parameter named GRAPH\_COMPLEXITY to limit this number. The GRAPH\_COMPLEXITY is the number of predecessors necessary to consider in graph generation.
- At the end of the speech signal, the letter graph is constructed by tracking back through the bookkeeping lists. For each letter obtained in the backtracking process, we have to calculate:
  - The letter boundary  $(t_{\text{initial}}, t_{\text{final}})$ .
  - The letter score  $(L_{\text{score}})$ .

One node in the graph is characterised by three parameters: letter, initial frame and final frame. In the graph generation, if we obtain several nodes with same letter, initial frame and final frame, these nodes are joined, maintaining the lowest letter score. In several conditions it is possible to relax these constraints and join nodes with same letter but with small variations in the initial and final frames. An example of a letter graph is shown in Fig. 7.

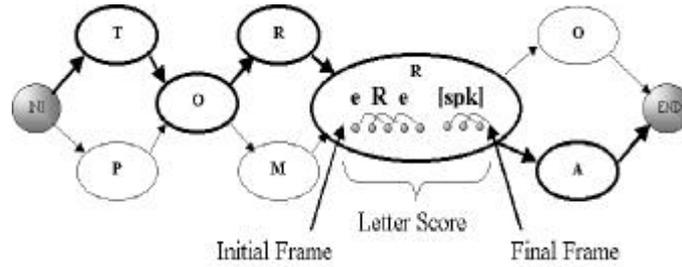


Fig. 7: Letter Graph generated for the spelled name TORRA with a GRAPH\_COMPLEXITY=2.

There are three comments to the letter graph method that are important to remark upon.

- Following the refinements proposed by H. Ney for short words like articles and prepositions, the noise models are joined with the predecessor letter to form a node. In Fig. 7, we can see how the noise [spk] has been joined with the letter model R to form the node R.
- The noise models at the beginning of the graph (before T or P in our case) are joined to the first nodes (T or P).
- The silent models have not a special treatment because we consider them part of the letter model (section 4.2).

To generate the N-Best letter sequences we do not rerun the HMM recognizer with this structure, we order the letter sequences considering the letter score calculated in each node and the LM penalty between letters (in this case we consider the 2-gram LM). To improve the Name Recognition Rate using the N-best letter sequences, we follow the ideas shown in (Jouvet, 1993b). The first idea is use the N sequences to train the penalties of the DP algorithm, assuming that each of the N-best solutions provides a realistic example of recognition behaviour. And, considering that the N solutions are possible recognition answers, we apply the retrieval procedure to all of them to obtain the most probable answer.

The Letter Accuracy has been calculated considering the sequence that produced the best alignment (lowest cost) with any name from the dictionary using the DP algorithm. The results obtained are shown in table 6.

Table 6.

Letter Accuracy (LA), Name Recognition Rate (NRR) and Processing Time (PT) for the N-Best letter sequences experiments with the 2-gram LM with N=1,2,4 and 8.

|        | LA (%) | NRR(%) | PT(xRT) |
|--------|--------|--------|---------|
| 1-Best | 81.4   | 92.3   | 1.3     |
| 2-Best | 85.4   | 93.1   | 1.5     |
| 4-Best | 89.1   | 94.1   | 2.2     |
| 8-Best | 90.5   | 94.3   | 2.3     |

It is important to stress that when we increase the number of sequences considered, the Name Recognition Rate reaches saturation. The reason for this behaviour is that when we increase the number of letter sequences, the assumption that each of the N-best solutions provides a realistic example of recognition behaviour is false.

In our experiments, we have used a GRAPH\_COMPLEXITY of 2 for 1-Best and 2-Best and a GRAPH\_COMPLEXITY of 3 for 4-Best and 8-Best. We use a higher value in the second case to guarantee the possibility of obtaining N different letter sequences. In these experiments, we have obtained 16.3 and 19.2 average number of nodes per graph respectively. Considering

that the average number of letters for the names in the testing set is 7.6 letters, we can conclude that the HMM models are performing quite well, permitting us to join a significant number of nodes with same letter, initial frame and final frame (even under these strict constraints).

Shown the advantage of using the letter graph for N-Best generation we consider the possibility of incorporating the 3-gram LM into the graph. The trigram probability for the transition depends on two letters before the letter being considered. Since there can be more than one such predecessor in the letter graph, the grammar probability for the node under consideration is not uniquely determined. The difficulty is easily solved by the usual method of replicating a node for each distinct predecessor, i.e. creating distinct grammar states in the letter graph. We illustrate this process with an example in the Fig. 8.

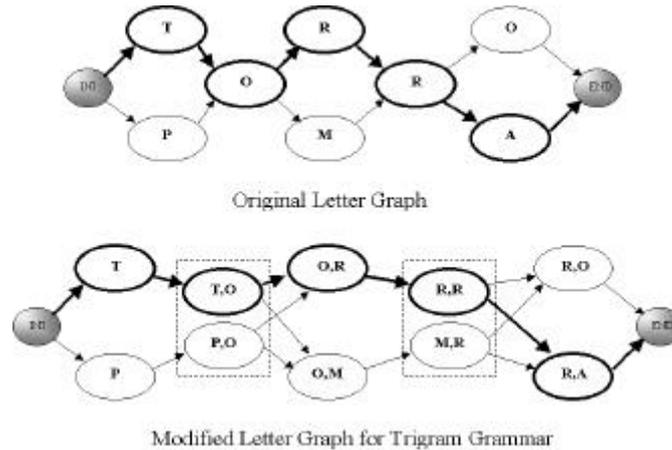


Fig. 8: Modification of the letter graph for Trigram Grammar.

The modification of the letter graph is straightforward:

1. If a node (i.e. O) has  $n$  distinct predecessors (T and P) in the original letter graph, it is replicated  $n$  times in the new letter graph: (T,O) and (P,O). The first component of the label identifies a predecessor letter. Instances of such replication where  $n > 1$  are marked by dashed rectangles in Fig. 8.
2. If there is an edge from O to R or to M in the original letter graph, the new letter graph has an edge from every copy of O to (O,R) and (O,M).
3. The acoustic score component of all node copies is the same than the acoustic score of the original node.

In table 7, we can see the results obtained incorporating the 3-gram language model and considering N-Best letter sequences.

Table 7.

Letter Accuracy (LA), Name Recognition Rate (NRR) and Processing Time (PT) for the N-Best letter sequences experiments with the 3-gram LM with N=1,2,4 and 8.

|        | LA (%) | NRR(%) | PT(xRT) |
|--------|--------|--------|---------|
| 1-Best | 85.4   | 93.4   | 2.0     |
| 2-Best | 88.7   | 94.2   | 2.1     |
| 4-Best | 90.8   | 94.6   | 3.0     |
| 8-Best | 92.3   | 94.9   | 3.0     |

We can see again how the incorporating of a 3-gram LM has increased considerably the Letter Accuracy but the increment in the Name Recognition Rate is small. Another effect we

can see is that the Name Recognition Rate saturates when the number of letter sequences considered increases. The incorporation of the 3-gram LM in the letter graph has increased the average number of nodes in the graph: 25.1 and 36.1 for GRAPH\_COMPLEXITY 2 and 3 respectively. This fact, plus the need to consider the graph modification algorithm has produced the Processing Time to increase a little.

We can conclude that Letter Graph Generation is a very good approach to incorporate complex LMs into the recognizer and to calculate the N-Best letter sequences. Observing the Name Recognition Rate and the Time Consumption for the alternatives presented during section 4 for the hypothesis stage, we decided to consider the Letter Graph Generation to calculate the 2-Best letter sequences with the 3-gram LM.

#### 4.6. Analysis of the most confusable letter sets.

For the final configuration of the hypothesis stage, we present in table 8 the percentage of substitutions (Sub), deletions (Del) and insertions (Ins) on the most confusable sets. These percentages has been calculated with the equations (2), (3) and (4).

$$Sub(\%) = \frac{N_s}{N_T} \times 100 \quad (2)$$

$$Del(\%) = \frac{N_D}{N_T} \times 100 \quad (3)$$

$$Ins(\%) = \frac{N_I}{N_T} \times 100 \quad (4)$$

where.

- $N_s$ : number of letters, within the set under analysis, substituted by any other letter, belonging or not to the same confusable set.
- $N_D$ : number of letters, within the set under analysis, deleted from the reference letter sequence.
- $N_I$ : number of letters, within the set under analysis, inserted in the letter sequence.
- $N_T$ : total number of letters in the reference letter sequence, belonging to the set under analysis.

Table 8.  
Substitutions (Sub), Deletions (Del) and Insertions (Ins) on the most confusable sets.

|         | Sub(%) | Del(%) | Ins(%) |
|---------|--------|--------|--------|
| Total   | 8.5    | 1.6    | 1.2    |
| E-set   | 15.5   | 0.6    | 1.9    |
| ExE-set | 11.7   | 1.9    | 2.1    |
| [F,S]   | 7.7    | 2.9    | 2.4    |
| [L, LL] | 12.5   | 1.9    | 3.1    |
| [M,N,Ñ] | 12.1   | 0.6    | 2.3    |
| [K, A]  | 1.4    | 1.6    | 0.5    |
| [Q, U]  | 9.8    | 1.3    | 1.4    |

The highest percentage of substitutions obtained is on the E-set and in the ExE-set, (specially for [L, LL] and [M,N,Ñ]) and the lowest is for the sets [K, A] and [Q, U]. In these cases although the acoustic confusion is high, the first letter is a consonant and the second is a vowel, so the language model can discriminate them more easily.

## 5. Verification stage

As we mentioned in Section 2, in the verification stage a dynamic grammar is built with the M-Best candidates provided by the hypothesis stage (alignment module) and the HMM recognizer is invoked again with this constrained grammar. The problem now is to calculate the parameter M, the number of candidates to build the constrained grammar.

When we work with hypothesis-verification based systems it is necessary to study carefully the number of candidates M to consider. We represent the Name Recognition Rate depending on M with the equation 5.

$$NRR(M) = P_H(M) \times P_V(M) \quad (5)$$

Where:

- $P_H(M)$  is the probability of obtaining the name spelled into the M candidates after the hypothesis stage.
- $P_V(M)$  is the rate of names recognized in first position after the verification stage, considering only the cases where this name was one of the M candidates.

In the Fig. 9, we can see  $P_H(M)$  and  $P_V(M)$  obtained for the 1,000 name dictionary.

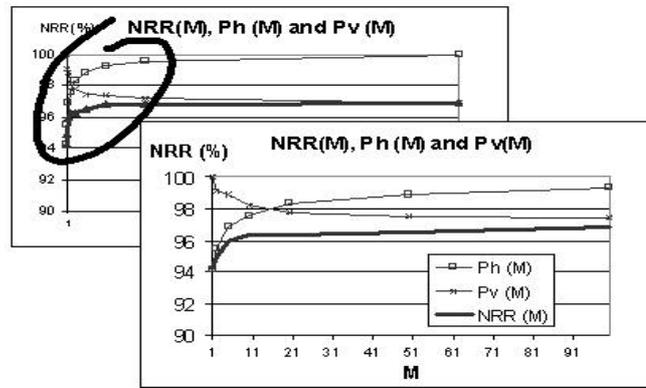


Fig. 9: Name Recognition Rate (NRR),  $P_H(M)$  and  $P_V(M)$  for the 1,000 names dictionary.

$P_H(M)$  and  $P_V(M)$  have opposite behaviours. When we consider higher M values, the probability of including the spelled name into the M candidates ( $P_H$ ) increases but the recognition rate in the verification step ( $P_V$ ) decreases because the number of names to be considered into the constrained grammar is higher.  $P_V(M)$  decreases quickly with small M values and stay almost constant for large M values. The reason of this behaviour is as follow: the hypothesis step provides the most similar names to the spelled letter sequence, so the confusion between these names is very high. When we include one name with small M, we incorporate a lot of confusion in the constrained grammar and  $P_V(M)$  decreases quickly. When we consider more names with large M, these names are so different from the spelled letter sequence and we do not add more confusion, maintaining  $P_V(M)$  almost constant. When we use a hypothesis-verification approach it is important to analyze  $P_H(M)$  and  $P_V(M)$  in order to obtain an M value with a good compromise between Name Recognition Rate and Time Consumption.

For 5,000 and 10,000 directories, the representation for  $P_H(M)$  and  $P_V(M)$  was similar to the 1,000 name directory case (Fig. 9). We calculated the best M as a compromise between Time Consumption and Name Recognition Rate using the development set. We present the final results with the testing set in table 9. In this table, we present the Hypothesis Name Recognition Rate (i.e. the recognition rate obtained after the DP alignment considering only the hypothesis step, without the verification step). The Hypothesis-Verification Name Recognition Rate is the recognition rate obtained considering the whole system.

Table 9.

Final results for dictionaries of 1,000, 5,000 and 10,000 names.

| Size of the dictionary | Hypothesis Name Recognition Rate | Hypothesis-Verification Name Recognition Rate | M  | Time (xRT) |
|------------------------|----------------------------------|---|----|------------|
| 1,000 (0.2)            | 94.2%                            | 96.3%   | 10 | 2.8        |
| 5,000 (0.5)            | 88.7%                            | 92.8%   | 20 | 3.4        |
| 10,000 (0.9)           | 86.2%                            | 90.3%   | 50 | 4.7        |

We can analyze and compare here previous results reported in Table I in (Junqua, 1997) for a similar task for English. In Table 10, we present the result for  $M = 20$ .

Table 10.

Final results for dictionaries of 1,000, 5,000 and 10,000 names considering  $M=20$ .

| Size of the dictionary | Hypothesis-Verification Name Recognition Rate | M  | Time (xRT) |
|------------------------|---|----|------------|
| 1,000 (0.2)            | 96.3%   | 20 | 3.2        |
| 5,000 (0.5)            | 92.8%   | 20 | 3.4        |
| 10,000 (0.9)           | 90.0%   | 20 | 3.5        |

Table 11.

Final results reported in (Junqua, 1997).

| Size of the dictionary | Hypothesis-Verification Name Recognition Rate | M  |
|------------------------|---|----|
| 491 (0.07)             | 98.4%   | 20 |
| 3,388 (0.50)           | 95.3%   | 20 |
| 21,877 (1.80)          | 90.4%   | 20 |

In (Junqua, 1997), the authors considered a database with 4,000 calls where people were asked to spell their first and last names, with and without pauses. More than 1,200 different calls were selected for training, 558 calls for validation and 491 calls for testing. All calls selected were last names produced without pauses. Moreover, none of the calls contained extraneous speech, line noise or speech related effects such as lipsmach or breath noises.

Our results may appear slightly worse but the task cannot be directly compared. Take into account that we consider files containing all kinds of noises. As we commented in section 4.3, in Spanish there is a significant relationship between pronunciation and spelling and people are not used to spelling for clarification. This fact causes the generation of false beginnings, doubts, filled pauses, mistakes and significant differences in the speaking rate. We have evaluated these differences in the testing set and we have obtained an average speaking rate of 1.1 l/s (letters per second) with a Typical Deviation of 0.24 l/s (Variance 0.06). The minimum rate observed was 0.4 l/s and the maximum was 2.1 l/s. The average confusion for each dictionary is indicated in parentheses.

A Real-time version of the system has been implemented on a Pentium III 600 Mhz with 256 Mb of RAM, working over the telephone network.

## 6. Field evaluation

The spelled name recognizer has been included in a Directory Assistance Service for Spanish with Large-Vocabulary Recognition over the Telephone. The system has been developed within the EU project IDAS (LE4-8315) (Lehtinen, 2000). Some of its characteristics follow:

- Representative database with about one million registers.

- 4 different vocabularies: cities, first names, surnames and company names. The vocabularies of cities, first names and company names have 1,000 words and the surnames uses 10,000 words.
- It provides telephone numbers: private and company numbers.
- Operator fallback if the recognition module fails.

In the case of a company number, the systems asks the user the city and the company name and for the case of private telephone numbers the city, surname and first name. We have considered a 1.000 words dictionary for city and company names and a 10.000 word dictionary for surnames. For first names, we have used a 1.000 words dictionary but, when the city and the surnames have been recognized correctly, the active first name dictionary is reduced to less than 50 names. The system presents the user with the first and second candidates from the Names Recognizer. If none of them is confirmed, the system asks the user to spell the name by invoking the Spelled Name Recognizer, as the final option before the operator intervention. In this last case, only the first candidate is presented to the user for confirmation. In this service we have not implemented any automatic method for out of vocabulary name detection. If one of the names cannot be recognized, the call has to be completed by an operator, who types the right name and makes the call to process. The user does not talk to the operator directly. This way the call duration can be shorter and the operator can manage several calls at the same time.

For dealing with possible long pauses between letters it has been necessary to modify the end point detector relaxing the time constraints for the end detection. This solution increases the response time a little bit but it guarantees less than 5% of utterances truncated.

Over a two-month period, a total of 600 calls were collected with 30 students from the University. 50% of the calls asked for company telephone numbers and 50% for private ones. The calls were recorded and the results were analyzed afterwards. The results of this evaluation for both Name Recognizer and Spelled Name Recognizer are set out in Table 12.

Table 12.  
Results for the field evaluation.

| Size of the dictionary | Name Recognizer       |                       | Spelled Name Recognizer | Global |
|------------------------|-----------------------|-----------------------|-------------------------|--------|
|                        | 1 <sup>st</sup> Cand. | 2 <sup>nd</sup> Cand. |                         |        |
| 1,000 (0.3)            | 62.2%                 | 8.1%                  | 15.7% (52.7%)           | 86.0%  |
| 10,000 (1.1)           | 32.7%                 | 7.5%                  | 21.7% (36.9%)           | 61.9%  |

The results presented correspond to the percentage of times that the system obtained the right name as first or second candidate from the Name Recognizer and as first candidate from the Spelled Name Recognizer. In parentheses we show the spelled name recognition rate, evaluated only with the cases where the Name Recognizer failed. For the 1,000 case, we present the average result over the different dictionaries (cities and companies) and for the 10,000 we considered the surnames dictionary. There is a significant degradation in performance because the dictionary confusions are higher and the Spelled Name Recognizer is working under difficult conditions, i.e. this recognizer is invoked only when the Name Recognizer failed. This means that there could be a significant background noise, the speech signals could have low energy or the user could be unused to talking to automatic systems.

In this evaluation, 39.4% of the calls were served automatically without the Spelled Name Recognizer, 19.3% were served automatically using the Spelled Name Recognizer and for the rest (41.3%), it was necessary the operator intervention<sup>2</sup>. Because of this, we can conclude that, although there is a significant degradation compared to laboratory experiments, the Spelled Name Recognizer permitted us to increase from 39.4% to 58.7% (39.4%+19.3%) the percentage of calls completed fully automatic.

## 7. Conclusions

In this paper, we present a hypothesis-verification approach for continuously spelled name recognition over the telephone and we give a detailed description of the spelling task for Spanish. We analyze different alternatives for the hypothesis step. We propose the use of a new HMM topology with contextual silences to model the pauses between letters improving by 6.6 points the Letter Accuracy compared with a single silence model approach. Including noise models in the search space increases the Letter Accuracy by 4.5 points (from 76.2% to 80.7%) and obtains a 92.0% Name Recognition Rate in this step. Modelling these noises is very important in a Spelled Name Recognizer when the user is not used to spelling.

We incorporate N-gram (2-gram and 3-gram) LMs into the decoding process. This grammar was generated from the directory of the task. We describe an efficient way to consider the noise models in the search space generated for the 3-gram, obtaining for this case a Letter Accuracy of 89.0% and a Name Recognition Rate of 93.2%.

We validate for this task that the letter graph (based on the word-graph proposed by H. Ney) is an efficient way to incorporate N-gram in the decoder process and to calculate the N-best letter sequences. Considering the Letter Graph Generation, we propose a configuration for the hypothesis step incorporating the 3-gram language model and calculating the 2-best letter sequences. In this situation, we obtain a Letter Accuracy of 88.1% and a Name Recognition Rate of 94.2% considering the 1,000 names dictionary.

For the verification step we consider a dynamic grammar built with the M-Best candidates provided by the hypothesis step. We describe the analysis to calculate a M with a good compromise between Name Recognition Rate and Time Consumption and we evaluate the whole system for 1,000, 5,000 and 10,000 names dictionaries obtaining, 96.3%, 92.8% and 90.3% recognition rates respectively. These results are comparable to systems developed for other languages.

Finally, we demonstrate the utility of incorporating a Spelled Name Recognizer in a Directory Assistance Service over the telephone. The field evaluation shows that the percentage of calls automatically serviced increased from 39.4% to 58.7%.

## 8. Acknowledgments

The authors are very grateful to Rubén Pacheco for his help in implementing the algorithm for training the HMMs. The authors wish to thank to all GTH Speech Technology Group at Technology University of Madrid for their contribution to this work and to the anonymous reviewers that helped with their comments to improve the quality and clarity of the presentation.

## Appendix A. Speaker Independent experiments

Analyzing in detail the description of the database (section 3), we can see that the 1000 random letter sequences (1 sequence from each of the 1000 speakers) are always part of the training set. Consequently for each of the 6-Round Robin training, data from each test speaker

---

<sup>2</sup> Note that one call is completed automatically when all the items have been recognized correctly. Every call consists of three items (private telephone numbers) or two items (company telephone numbers). The percentage of calls automatically served could be calculated multiplying the recognition rates for each item. In table 12, we only have the average results for dictionaries with the same size but not the results detailed for each dictionary. Because of this, the percentages of calls automatically served have to be obtained from the analysis of the calls and not by multiplying the recognition rates.

(and corresponding environment condition) are also present in training set. To analyze its influence, we have repeated the experiments for the sections 4.1, 4.2 and 4.3 removing the random letter sequences from the speakers contained in the development and final testing sets (300 audio files). These experiments include the baseline system, considering contextual silence models and incorporating noise models respectively. The results are presented in table 13.

Table 13.

Results for the HMM topology with contextual silences (HMM-CS) and the 4 noise models (N-HMMs) removing from the HMM training set data from the speakers contained in the development and final testing sets (average results for the 6-Round Robin training): Percentage of Substitutions (Sub), Insertions (Ins) and Deletions (Del), Letter Accuracy (LA), Name Accuracy (NA). Name Recognition Rate (NRR) and Processing Time (PT), consumed by the whole hypothesis stage (considering the 1,000 dictionary), are shown.

|                     | HMM recognizer |        |        |       |       | Hypothesis stage |         |
|---------------------|----------------|--------|--------|-------|-------|------------------|---------|
|                     | Sub(%)         | Ins(%) | Del(%) | LA(%) | NA(%) | NRR(%)           | PT(xRT) |
| Baseline            | 20.8           | 6.7    | 3.8    | 68.7  | 20.8  | 83.1             | 0.9     |
| HMM-CS              | 17.7           | 4.4    | 2.4    | 75.5  | 27.2  | 88.6             | 1.2     |
| HMM-CS + Noise HMMs | 16.0           | 1.2    | 2.5    | 80.3  | 33.8  | 91.7             | 1.2     |

In this case, incorporating contextual silence models in the letter model has improved the Letter Accuracy 6.8 points (from 68.7% to 75.5%) obtaining a 88.6% Name Recognition Rate for the hypothesis stage.

Table 4 shows the results obtained without removing from the HMM training set the files from speakers contained in the development and final testing sets.

When removing the files, the Letter Accuracy and the Name Recognition Rate have decreased a little. This decrement has been produced mainly because of the reduction in the number of audio files to train the HMMs (1,800 instead of 2,100) but not because of removing data from development and testing speakers. To validate this idea, we have conducted the following experiments including the contextual silences (not the noise models): considering only a 1-Round Robin, we have split the development and testing sets in 3 sub-sets and we have trained independent HMMs to recognize each sub-set. Removing the data of the development and testing speakers, we have 2,000 audio files to train every HMMs. The final results are the average results obtained in the 3 sub-sets. In a same way, we have split the development and testing sets in 10 sub-sets, considering in this case 2,070 audio files to train every HMMs. The results are shown in table 14.

Table 14.

Results for the HMM topology with contextual silences (HMM-CS) removing 300, 100 and 30 audio files respectively from the HMMs training set (considering only a 1-Round Robin)<sup>3</sup>:

|  | LA(%) | NRR(%) |
|--|-------|--------|
| Considering all data to train the HMMs (2,100)                   | 75.6  | 88.5   |
| Removing 300 files from development and testing speakers (1,800) | 74,8  | 87.5   |
| Removing 100 files from development and testing speakers (2,000) | 75,5  | 88,2   |
| Removing 30 files from development and testing speakers (2,070)  | 75,6  | 88,4   |

As we can see when we reduce the number of files removed from the training set the results improve reaching almost the same results considering all data to train the HMMs. In this

<sup>3</sup> Note that the data of this table are different from the ones in tables 13 and 4 because this experiments (in contrast to the experiments of tables 13 and 4) has been done using only 1-Round Robin data.

case, when testing the recognizer with a file, there is only one file (out of 2,100) from the same speaker for HMMs training, so its influence is irrelevant.

In this Appendix, we have not repeated the experiments when considering the language models because in these cases, the decrement in the acoustic model quality is less significant.

## References

- Bauer, J.G., Junkawitsch, J., 1999. Accurate recognition of city names with spelling as a fall back strategy. In Proc. EUROSPEECH. pp. 263-266.
- Brown, P.F., 1987. The Acoustic-modelling problem in automatic speech recognition. Ph. D dissertation. Dept Comput Sci, Carnegie-Mellon Univ., Pittsburgh, PA.
- Cole, R.A., Stern. R. M., and Lasry, M. J., 1986. Performing fine phonetic distinctions: Templates vs. Features, in Variability and Invariance in Speech Processes, J. S. Perkell and D.H. Klatt, Eds. Hillsdale, NJ: Lawrence Erlbaum, pp-325-345.
- Cole, R.A., Fanty, M., Gopalakrishnan, M., and Janssen, R.D.T., 1991a. Speaker-independent name retrieval from spellings using a data base of 50,000 name. in Proc. ICASSP, pp 325-328.
- Cole, R.A., Roginski. K. and Fanty, M. 1991b. English alphabet recognition with telephone speech. In Proc. EUROSPEECH, pp. 479-482.
- Euler, S.A. Juang, B-H, Lee, C-H. and Soong, F.K., 1990. Statistical segmentation and word modeling techniques in isolated word recognition. In Proc. ICASSP, pp- 745-748.
- Fanty. M. and Cole, R.A. 1990. Spoken letter recognition. In Proc. Neural Inform. Processing Syst. Conf. Nov. pp 220-226.
- Fissore, L., Laface, P., Micca, G., and Pieraccini, R., 1989. Lexical Access to large vocabularies for speech recognition. IEEE Transactions and Acoustics, Speech and Signal Processing. Vol 17 No 8, pp. 1197-1213.
- Hanel, S., Jouviet, D., 2000. Detecting the end of spellings using statistics on recognized letter sequences for spelled names recognition. In Proc. ICASSP. pp. 1755-1758.
- Hermansky, H., Morgan. N., Bayya A., Kohn. P., 1991. Compensation for the effect of the communication channel in auditory-like analysis of speech (RASTA-PLP). In Proc. EUROSPEECH. pp. 1367-1370.
- Hild. H., and Waibel. A., 1993. Speaker-independent connected letter recognition with a multistate time delay neural network. In Proc. EUROSPEECH. pp 1481-1484.
- Hild. H., and Waibel. A., 1997. Recognition of spelled names over the telephone. In Proc. EUROSPEECH. pp 346-349.
- Jouviet. D., Lainé. A., Monné. J., and Gagnoulet. C., 1993a. Speaker independent spelling recognition over the telephone. In Proc. ICASSP, pp. II.235-II.238.
- Jouviet. D., Lokbani. M.N., and Monné. J., 1993b. Application of the N-best solutions algorithm to speaker independent spelling recognition over the telephone. In Proc. EUROSPEECH, pp. 2081-2084.
- Jouviet. D., Monné, J., 1999. Recognition of spelled names over the telephone and rejection of data out of the spelling lexicon. In Proc. EUROSPEECH. pp. 283-286.
- Junqua, J.C., 1991. A two pass hybrid system using a low dimensional auditory model for speaker-independent isolated-word recognition. Speech Communications, vol. 10 pp 33-44.
- Junqua, J.C., Valente, S., Fohr, D., Mari, J.F., 1995. An N-Best strategy, dynamic grammars and selectively trained neural networks for real-time recognition of continuously spelled names over the telephone. In Proc. ICASSP, pp 852-855.
- Junqua, J.C., 1997. SmarTspell™ : A Multipass Recognition System for Name Retrieval over the Telephone. IEEE Trans. On Speech and Audio Processing, Vol. 5, No. 2, March.
- Kaspar. B., Fries. G., Schuhmacher. and Wirth. A., 1995. FAUST--a directory assistance demonstrator. In Proc. EUROSPEECH, pp 1161-1164.

- Kuroiwa, S., Naito, M., Yamamoto, S., and Higuchi, N., 1999. Robust speech detection method for telephone speech recognition system. *Speech Communications*, vol. 27 No 2, pp 135-148.
- Lamel, L., Rosset, S., Gauvain, J.L., Bennacef, S., Garnier-Rizet, H., Prouts, B., 2000. The LIMSI ARISE system. *Speech Communications*. Vol 31, No 4 pp 339-355.
- Lehtinen, G., Safra, S., Gauger, M., Cochard, J.L., Kaspar, B., Hennecke, H., Pardo, J.M., Códoba, R., San-Segundo, R., Tsopanoglou, A., Louloudis, D., Mantakas, M., 2000. IDAS : Interactive Directory Assistance Service. VOTS-2000 Workshop, Belgium.
- Linde, Y., Buzo, A., Gray, R.M., 1980. Vector quantization in speech coding. *IEEE Trans. Comm.*, Vol 28, pp 84-95.
- Loizou, P., Mekkoth, A., and Spanias, A., 1995. Telephone alphabet recognition for name-retrieval applications. In *Proc. ICSPAT*, pp 2014-2018.
- Loizou, P., and Spanias, A., 1996. High performance alphabet recognition. *IEEE Trans. On Speech and Audio Processing*, Vol. 4, No. 6.
- Moreno, A. *SpeechDat* [cd-rom]. Ver. 1.0. [Barcelona]: Universitat Politècnica de Catalunya <<http://www.upc.es/castella/recerca/recerca.htm>>, c1997. 4 cd-roms. (Spanish Fixed Network Speech Corpus).
- Mitchell, G., Setlur, A.R., 1999. Improved spelling recognition using a tree-based fast lexical match. In *Proc. ICASSP*, pp. 597-600.
- Ney, H., 1994. A word graph algorithm for Large Vocabulary Continuous speech recognition. In *Proc. ICSLP*. pp 1355-1358.
- Ney, H., and Ortmanns, S., 1999. Dynamic programming search for continuous speech recognition. in the *IEEE Signal Processing Magazine*, Vol 16 No 5 pp 64-83.
- Ravishankar, M.K., 1996. Efficient Algorithms for Speech Recognition. Unpublished PhD Dissertation, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, May 15.
- Roginski, K., 1991. A neural network phonetic classifier for telephone speech. Master thesis, Oregon Grad. Inst., Portland, OR.
- Schrâmm, H., Rueber, B., and Kellner, A., 2000. Strategies for name recognition in automatic directory assistance systems. *Speech Communications*. Vol 31, No 4 pp. 329-338.
- Thiele, F., Rueber, B., Klakow, D., 2000. Long range language models for free spelling recognition. In *Proc. ICASSP*, pp. 1715-1718.
- Weiss, N.A., Hasset, M.J., 1993. *Introductory Statistics*, 3rd ed., Addison-Wesley, Reading, MA, pp. 407-408.